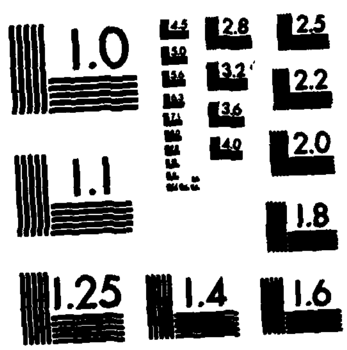
AD-A125 010 A COMPARISON OF CUBE TYPE AND DATA MANIPULATOR TYPE NETWORK(U) PURDUE UNIV LAFAYETTE IN SCHOOL OF ELECTRICAL ENGINEERING R J MCMILLEN ET AL. OCT 82 AFOSR-TR-82-1096 AFOSR-78-3581 F/G 9/5 1/1

UNCLASSIFIED NL

END

FILMED

DTIC



SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) UNCLASSIFIED

(2)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
AFOSR-TR- 82-1096	A125010	

4. TITLE (and Subtitle)

A COMPARISON OF CUBE TYPE AND DATA MANIPULATOR TYPE NETWORK

5. TYPE OF REPORT & PERIOD COVERED

Reprint

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

Robert J McMillen and Howard Jay Siegal

8. CONTRACT OR GRANT NUMBER(s)

AFOSR-78-3581

9. PERFORMING ORGANIZATION NAME AND ADDRESS

School of Engineering
Purdue University
West Lafayette, Indiana 47907

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

61102F 2304/A2

11. CONTROLLING OFFICE NAME AND ADDRESS

AFOSR/NM
Building 410
Bolling AFB, DC 20332

12. REPORT DATE

October 1982

13. NUMBER OF PAGES

6

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

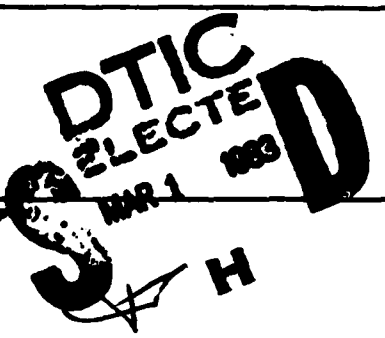
15. SECURITY CLASS. (of this report)

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release;
distribution unlimited.



17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Presented at the 3rd International Conference on Distributed Computing Systems, October 1982.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Teh interconnection of a large number of processors and other devices to form a parallel/distributed computing system is a research area receiving a great deal of attention. One method is to use a multistage network. This paper compares two classes of multistage networks by examining two representative networks: the Generalized Cube and the Augmented Data Manipulator. The two topologies are compared using a graph theoretic approach. By interpreting the graphical representations of the networks in different ways, different implementations result. The costs of the various implementations

-over-

DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

Item 20 continued: are compared taking VLSI considerations into account. Finally, the robustness of the different networks is measured and contrasted.

UNCLASSIFIED

Presented at the 3rd International Conference on Distributed Computing Systems, October 1982

A COMPARISON OF CUBE TYPE AND DATA MANIPULATOR TYPE NETWORKS

Robert J. McMillen and Howard Jay Siegel

School of Electrical Engineering
Purdue University
West Lafayette, IN 47907 USA

Abstract

The interconnection of a large number of processors and other devices to form a parallel/distributed computing system is a research area receiving a great deal of attention. One method is to use a multistage network. This paper compares two classes of multistage networks by examining two representative networks: the Generalized Cube and the Augmented Data Manipulator. The two topologies are compared using a graph theoretic approach. By interpreting the graphical representations of the networks in different ways, different implementations result. The costs of the various implementations are compared taking VLSI considerations into account. Finally, the robustness of the different networks is measured and contrasted.

very large scale integration

I. Introduction

The interconnection of a large number of processors and other devices to form a parallel/distributed computing system is a research area receiving a great deal of attention. Many different approaches to the interconnection method have been proposed and discussed including the use of buses [42], hierarchies of buses [39], direct links [10], single stage networks [16], multistage networks [7, 17, 27, 33], and crossbars [44]. An important aspect of this research is the evaluation and comparison of the proposed approaches [4, 13, 35, 40]. The conclusion most often reached is that the best scheme to use in a particular design highly depends upon the intended application, performance requirements, and cost constraints. Once a connection method is chosen (e.g. single stage network), a specific design must be decided upon and then implemented. During this phase of a system's specification, it is important for the designer to understand fully the differences and similarities between candidate designs. This paper is an investigation of two classes of multistage networks that have been considered for use in a number of systems. In particular, this work is part of a network evaluation study for the PASM [36] and PUMPS [8] systems.

The Generalized Cube and Augmented Data Manipulator (ADM) networks are defined in Section II. Their relation to other multistage networks described in the literature is also discussed. Using graph theory, the networks' topologies will be compared in Section III. In Section IV, two implementations resulting from two different graph interpretations will be examined to compare the cost of each network. Here, using VLSI chips is considered and costs are compared relative to the fraction of a stage that can be implemented on one chip. Finally, Section V will contain an analysis of the robustness each network exhibits.

This work was supported by the Air Force Systems Commands, USAF, under Grant No. AFOSR-78-3581, and the National Science Foundation under Grant No. ECS 80-19689. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

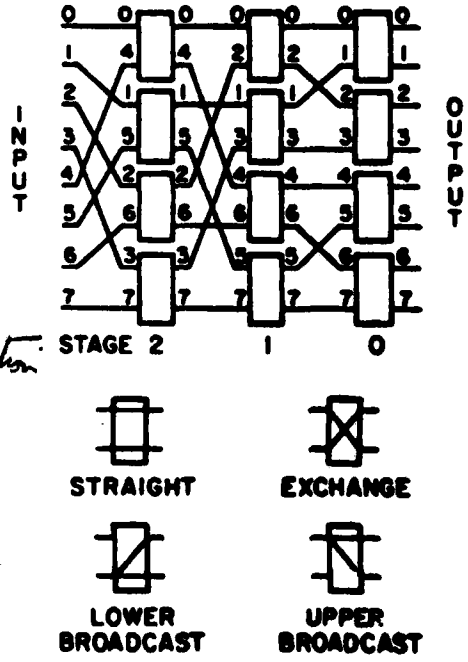


Figure 1: Generalized cube network for N=8. The four legitimate states of an interchange box are shown.

II. The Generalized Cube and ADM Networks

The Generalized Cube network is a multistage cube type network topology that was introduced as a standard for comparing network topologies [34]. Assume the network has N inputs and N outputs: in Fig. 1, N=8. The Generalized Cube topology has $n=\log_2 N$ stages, where each stage consists of a set of N lines connected to N/2 interchange boxes. Each interchange box is a two-input, two-output device. The labels of the input/output lines entering the upper and lower inputs of an interchange box serve as the labels for the upper and lower outputs, respectively. Each interchange box can be set to one of the four legitimate states shown [17].

The connections in this network are based on the cube interconnection functions [34]. Let $P=p_{n-1}\ldots p_1p_0$ be the binary representation of an arbitrary I/O line label. Then the n cube interconnection functions can be defined as:

$$cube_i(p_{n-1}\ldots p_1p_0) = p_{n-1}\ldots p_{i+1}\bar{p}_ip_{i-1}\ldots p_1p_0$$

where $0\le i<n$, $0\le P<N$, and $\bar{p}_i$ denotes the complement of p_i. This means that the $cube_i$ interconnection function

CH1808-8/82/0000/0614$00.75 © 1982 IEEE

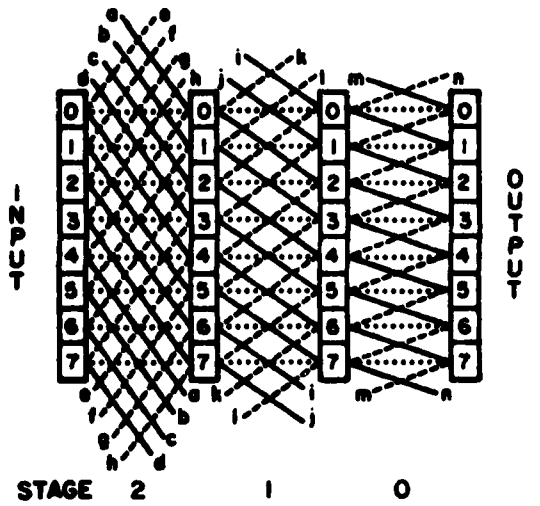


Figure 2: Augmented data manipulator network for N=8. (Lowercase letters represent end-around connections.)

connects P to cube$_i$(P), where cube$_i$(P) is the I/O line whose label differs from P in just the i^{th} bit position. Stage i of the Generalized Cube topology contains the cube interconnection function. That is, it pairs I/O lines that differ in the i^{th} bit position.

The ADM network is shown in Fig. 2 for N=8. It is based on Feng's data manipulator [12]. In this network, a *stage* consists of N *switching elements* or *nodes* and the 3N data paths that are connected to the inputs of a succeeding stage. Each node can connect one of its inputs to one or more of its outputs. At stage i of the ADM network, $0 \le i < n$, *the first output of node j is connected to the input of node* $(j-2^i) \bmod N$ of the next stage; the second output is connected to the input of node j; and the third output is connected to the input of node $(j+2^i) \bmod N$. Because $(j-2^{n-1})$ equals $(j+2^{n-1}) \bmod N$, there are actually only two distinct data paths instead of three from each node in stage n-1 (in the figure, stage 2). There is an additional set of N nodes at the output stage.

A number of systems have been proposed and/or built that use multistage networks [e.g. 5, 6, 19, 31, 36]. Among the networks that have been proposed are the ADM [33], baseline [43], binary n-cube [27], data manipulator [12], delta [26], Gamma [25], Generalized Cube [34], inverse ADM [23], omega [17], STARAN flip [7], and SW-banyan [15]. Studies have shown that the baseline, binary n-cube, Generalized Cube, omega, STARAN flip and SW-banyan (S=F=2) networks are all topologically equivalent [28, 32, 37, 43]. Differences between these networks are due to proposed control schemes, whether or not a broadcast capability is included, and the method used to number input and output ports. All of these networks belong to the general class of cube type networks. This in turn is in the class of banyan networks. They can also be considered delta networks since each can be controlled using one digit of a control vector (or number) per stage. Because of the similarities among these networks, a designer is not faced with choosing between seven different networks, rather the choice is whether or not to use a cube type network.

The data manipulator, ADM, IADM, and Gamma networks are topologically identical. The differences between these networks are the control scheme, order stages are transversed and switch complexity. The switches in each stage of the data manipulator are divided into two groups. Each group receives an independent set of control signals and all switches in a group respond identically. Each switching element of the ADM, IADM, and Gamma networks is controlled individually. The stages of the IADM and Gamma networks are traversed in an order opposite to that of the ADM and data manipulator. Also, the Gamma network's switching elements are 3x3 crossbars (as opposed to selecting one input at a time). None of the networks is a member of the banyan or delta classes. The capabilities of the Gamma network are a superset of the ADM and IADM networks. It has been shown in turn that their capabilities are a superset of all the cube type networks as well as the data manipulator network [37]. Data manipulator type networks, however, are more complex than cube type networks. For example, there are multiple paths between all nontrivial source/destination pairs (i.e. source address ≠ destination address). The redundancy provides a degree of fault tolerance. A common feature of all cube type networks is that there is exactly one path through the network for each source/destination pair. This property makes control schemes simple but any single failure of a link or switch will disallow the use of any path requiring the failed component.

Thus there exists the classic tradeoff between cost and performance when choosing between the two network types. In this paper, one representative network from each type will be compared: the Generalized Cube and the ADM. Both networks have the same number of input and output ports and individual switching element control. Routing tag schemes are available for the networks [17, 24, 33, 34], so it is assumed that they are used to implement network control.

Some aspects of the Generalized Cube and the ADM networks have been compared elsewhere. The ability of the ADM network to perform all the functions a Generalized Cube can was demonstrated in [37]. In [1], the total number of unique permutation connections each network can perform was compared. This paper is concerned with comparing cost and robustness or inherent fault tolerance. Cost is examined from two points of view. The first is the common method of counting links and switching nodes. In this case, graph theory with a consistent interpretation (two are possible) is used to insure a "fair" comparison. The second point of view is oriented toward VLSI considerations. Modules for each network requiring roughly the same number of pins are compared. The change in relative cost is also examined when as much as one whole stage is placed on one chip. Robustness is measured by calculating the average number of network inputs and outputs affected by the removal of a single link or switching element. The calculations are performed for both of the graph interpretations to be defined.

III. Graph Theory: A Common Ground for Comparing Networks

Graph theory has been used by Goke and Lipovski as the basis for defining a class of networks called banyans [15]. The graphs used to represent these networks consist of *nodes* connected by *undirected arcs*. By definition, in a banyan there is one and only one path from input to output [15]. In this paper there is no restriction on the number of paths from input to output.

It has been observed in [18] that the Generalized Cube network (Fig. 1) has the graphical representation shown in Fig. 3. This graph also represents an SW-banyan (with S=F=2). The graph can be interpreted a number of different ways. One is to treat each node (vertex) (a circle

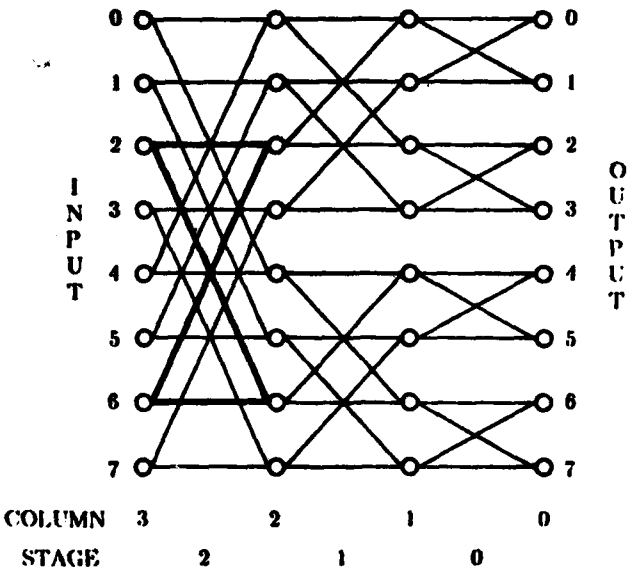


Figure 3: Graphical representation of the Generalized Cube network for N=8.

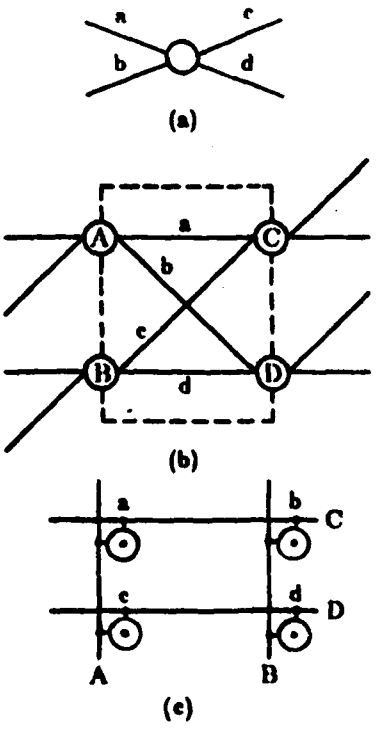


Figure 4: (a) A node from the graph representing the Generalized Cube network. When equated with a switch, input a *or* b can be connected to output c *or* d. (b) Four nodes from the graph. When the arcs, a, b, c, and d, are equated with switches, a 2x2 crossbar is obtained. (c) The components of a crossbar that correspond to the graph in (b).

in the figure) as a switch and each arc (edge) (a line in the figure) as a link. To model the network's behavior under this interpretation, the switch (node) shown in Fig. 4a should only connect one of the input links, a *or* b, to one of the output links, c *or* d. An implementation based on this interpretation, for an N input/output network, would consist of n+1 stages of N switches, with 2N lines between stages. The TRAC reconfigurable, multimicroprocessor system contains an SW-banyan constructed from switches of this type (but that have two incoming and three outgoing links, i.e. S=2 and F=3) [29]. A second interpretation is to treat the nodes as links and the arcs as forming interchange boxes. For example, the thickened lines in Fig. 3 can be considered to represent the interchange box with inputs 2 and 6 (compare this to Fig. 1). In this case the SW-banyan implementation would have the same structure as specified here for the cube (assuming a bidirectional network). This interpretation is illustrated in Figs. 4b and 4c. Each of the arcs labeled a through d in Fig. 4b acts as a crosspoint switch in Fig. 4c. When viewed this way, the portion of the graph within the dashed lines of Fig. 4b behaves as a 2x2 crossbar or interchange box. If a and d are "on," the straight setting is obtained; b and c on corresponds to exchange; a and b on corresponds to upper broadcast; and c and d on corresponds to lower broadcast. Conflict occurs if a and c or b and d are on at the same time. A third possible interpretation of the graph in Fig. 3 is to equate nodes with 2x2 interchange boxes and arcs with links. In that case, Fig. 3 would represent a size N=16 Generalized Cube network. This interpretation will not be discussed further in this paper.

The graphical representation of the ADM network (Fig. 2) is shown in Fig. 5. Since there are multiple paths from input to output, this is not a banyan graph. This

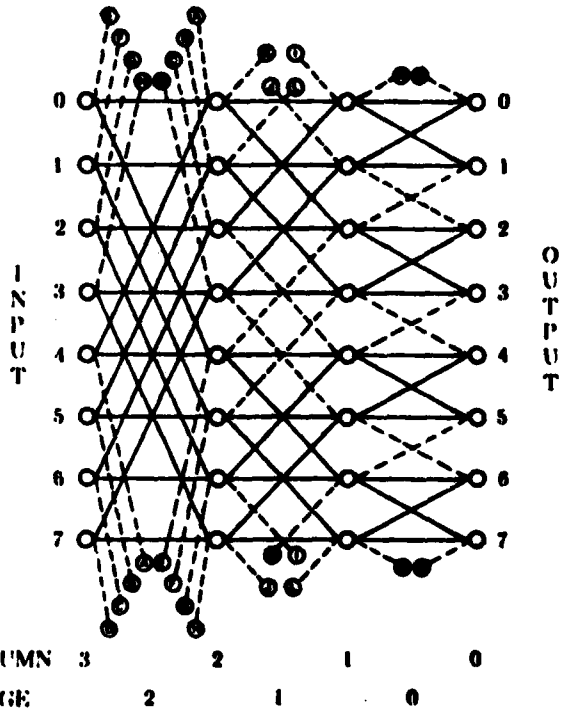


Figure 5: Graphical representation of the augmented data manipulator for N=8.

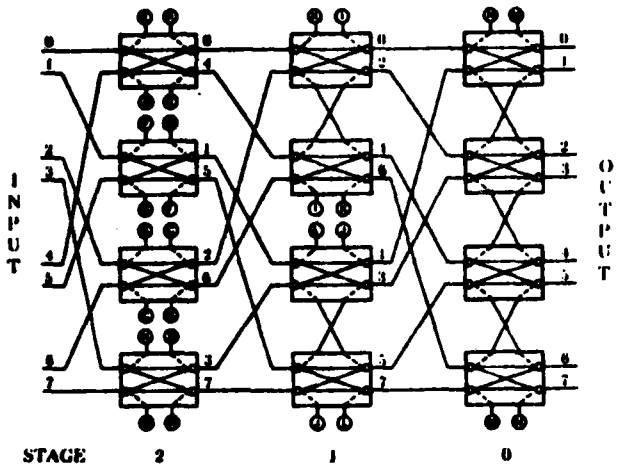


Figure 6: Implementation of the augmented data manipulator for N=8 when the graph of Fig. 5 is interpreted with arcs equated to switches.

graph can be obtained by adding the dashed lines shown to the graph in Fig. 3. When switches are equated with nodes, the network depiction in Fig. 2 is obtained. When switches are equated with arcs, the network looks like that shown in Fig. 6. In the figure, two nodes directly connected by a solid line between stages are represented by a single node in Fig. 5. Note that the labels on end-around connections in both Fig. 5 and Fig. 6 are attached to the same arcs (links) in the network. This second type of implementation is examined in [38], where LSI packaging of network building blocks is discussed.

Though the same ADM network is represented, Figs. 2 and 6 look rather different. Depending upon which representation is chosen, a comparison with the Generalized Cube in Fig. 1 could produce different conclusions. Comparing Figs. 1 and 2, one might conclude that, in addition to having an extra column of switches, the ADM has twice as many switching nodes and three times as many links as the Generalized Cube network. It would be easy to decide that the ADM network is considerably more expensive. On the other hand, comparing Figs. 1 and 6, it appears the only difference is N extra links that interconnect switches within each stage of the ADM network. The latter comparison is more accurate because the network depictions of Figs. 1 and 6 are based on the same interpretation of the networks' respective graphs. Thus when making comparisons, it is important to either compare graphical representations or consistent interpretations of those graphs. In the next section, the latter is done for both interpretations. This is so the resulting implementations can be compared as well.

IV. Cost Comparison

A. Introduction. The purpose of this section is to compare the cost of the Generalized Cube network to that of the ADM network. To do this, implementations of each network are examined. Since two basic implementations are possible for each network, to be fair, only implementations corresponding to the same graph interpretation are compared.

B. Hardware Realizations. There are two basic ways to implement multistage networks. They can be circuit switched or packet switched. In circuit switching, a complete path is established from input to output and must be held for the duration of the communication. Circuit switching is often used when processors are connected to the network inputs and memories are connected to the outputs. Designs for circuit switched interchange boxes have been discussed in [9, 22, 38]. In packet switching, messages are decomposed into packets which each make their way from stage to stage until the output is reached. This method is often used in configurations that connect processing element (*processor/memory pair*) *j* to input *j* and output *j* of a unidirectional network. Packet switched network switching element designs have been discussed in [11, 22, 41].

In the remainder of this paper, implementations will be discussed primarily in terms of packet switching. Circuit switched versions can be obtained by replacing any queues shown with busses. Other than this, remaining differences are in the control logic, however the logic is shown only at the block diagram level. Only key elements of the implementations to be discussed are included since many variations of the basic designs are possible. For more detail see [11, 22, 41].

C. Generalized Cube. Fig. 7 shows two designs for a Generalized Cube switching element. Fig. 7a results when switches are equated with nodes in the graph (this corresponds to Figs. 4a and 3). One of the two inputs is selected depending on the requests (if any) received by the (left half of the) control logic, which handles any needed arbitration. A single output link is shown, but it is to be

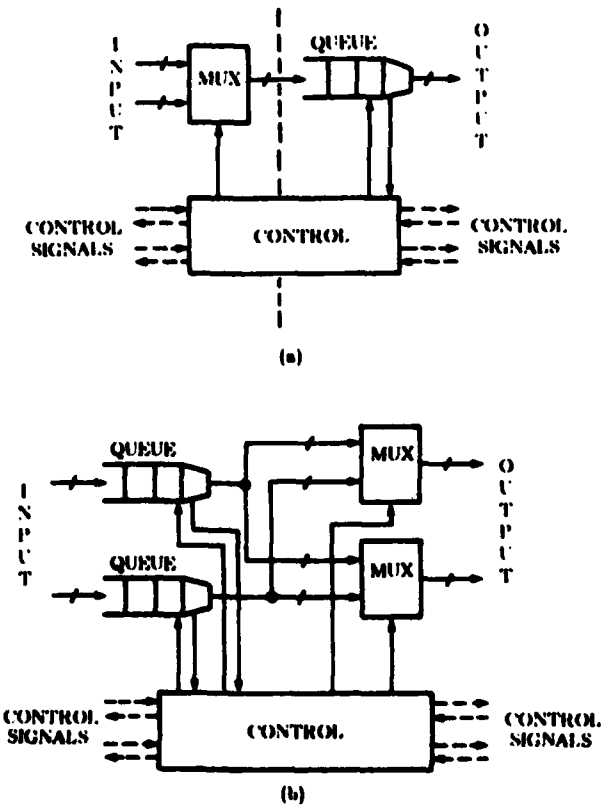


Figure 7: Implementation of Generalized Cube switches. (a) node = switch interpretation. (b) arc = switch interpretation.

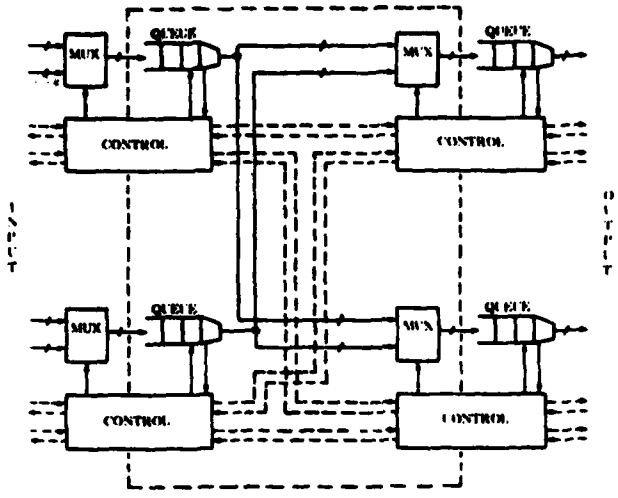


Figure 8: Four switches from Fig. 7a combined to form one switch (within dashed lines) equivalent to that in Fig. 7b.

connected to two other switches as shown in Fig. 8. A bit in the routing tag is examined by the control logic which then determines to which switch a request for access should be made. The (right half of the) control logic maintains the queue, interprets the routing tag, generates access requests, and receives grants for access requests. Switches that implement nodes in column 3 of Fig. 3 only contain hardware to the right of the dashed line in Fig. 7a. Switches that implement column 0 nodes only contain hardware to the left of the dashed line. A detailed design of this type is discussed in [29].

If arcs in the graph are equated with switches, then four arcs form a 2x2 crossbar or interchange box (see Figs. 4b, 4c, and 1). An implementation for this is shown in Fig. 7b. Here two input queues are required. As long as a given queue is not full, incoming packets for that queue will be accepted. Logic is required to handshake with other interchange boxes, maintain two queues, and interpret the routing tags at the head of each queue. This logic only interprets the tags in order to request the desired settings for the multiplexers. Logic associated with the multiplexers performs any necessary arbitration. It also makes appropriate requests of other interchange boxes once the multiplexers are set. Different protocols and design variations for this type of switching element are discussed in [22]. The performance of networks implemented with these interchange boxes has been studied in [11, 21].

The equivalence of two networks implemented with the two kinds of switching nodes is illustrated in Fig. 8. Four of the switching elements shown in Fig. 7a are connected as prescribed by the graph in Fig. 3. It can be seen that the hardware within the dashed lines is identical to that shown for the interchange box in Fig. 7b. The handshaking lines (directed dashed lines) shown connecting control units are equivalent to internal connections between the tag interpretation and queue control logic and the arbitration and output request logic in the control unit of Fig. 7b. It is thus apparent that the same total amount of hardware is required for either implementation, but that the two graph interpretations lead to different network building blocks or packagings for the components.

D. Augmented Data Manipulator. Two implementations for the ADM network are shown in Fig. 9. Fig. 9a results

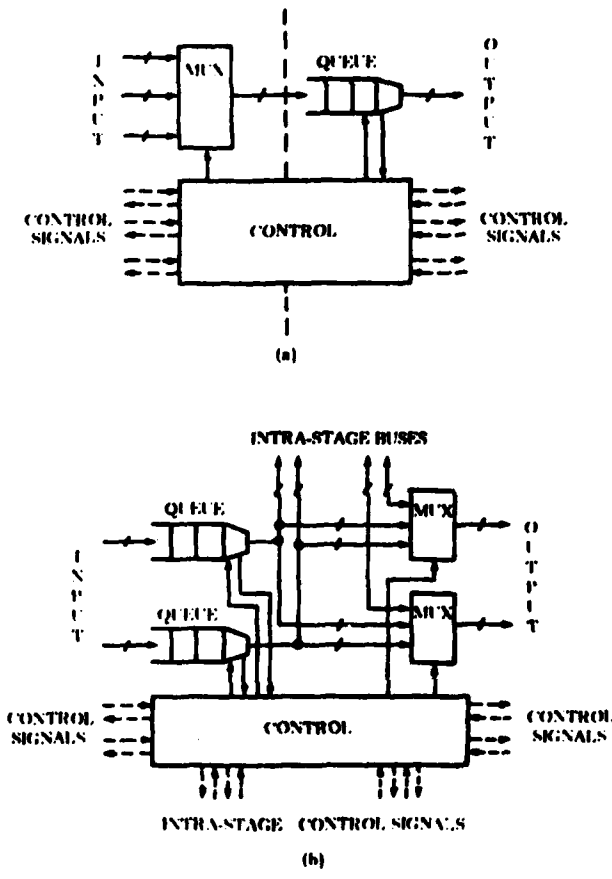


Figure 9: Implementation of augmented data manipulator switches. (a) node = switch interpretation. (b) arc = switch interpretation.

from equating the nodes of Fig. 5 with switches. In this design, the multiplexer selects from among three input links and the output link is connected to three other switches. The control signals shown on the output side in Fig. 9a are used to determine which of the switches is to read the data from the output link. A broadcast is performed by selecting more than one switch. The basic routing tag scheme for the ADM network [24] requires the routing tag logic to examine two bits, so it is slightly more complex than that required in the Generalized Cube. As with the Generalized Cube, the switches implementing nodes in columns 0 and 3 of Fig. 5 only require the logic to the left and right, respectively, of the dashed line in Fig. 9a.

If arcs are equated with switches, an implementation similar to the interchange box is obtained as shown in Fig. 9b. Here, however, the outputs from the queues must be connected to multiplexers in two other switching elements (as shown in Fig. 6) via *intra-stage busses*. Similarly, the two multiplexers shown here must accept connections from the queues of two other switching elements. Two control signals must also accompany each of the intra-stage busses.

E. Comparison. An approximate cost comparison between the Generalized Cube and the ADM network can be made by comparing their respective switching elements.

Since the choice is arbitrary, Figs. 7a and 9a will be compared. Both require a single queue. If the cost of the queue and its associated control logic dominates the cost of the switching element, then the ADM switch will cost only slightly more than a Generalized Cube switch. On the other hand, for a circuit switched implementation, the multiplexer and control logic in an ADM switching element will cost about 50% more than that required in a Generalized Cube switching element.

The perspective changes somewhat when implementing these four designs in VLSI is considered. Input/Output (I/O) requirements and logic/pin ratio become important considerations. For constructing a Generalized Cube network, the interchange box in Fig. 7b is a better choice than the switch in Fig. 7a. The interchange box (Fig. 7b) has approximately 33% more pins but approximately 100% more logic than the switch (Fig. 7a). For the ADM network, the logic/pin ratio is nearly the same for both of the designs in Fig. 9. The design in Fig. 9b has approximately twice as many pins and twice as much logic as that shown in Fig. 9a. The extra links that give the ADM network its superior capabilities over the Generalized Cube require a larger number of pins on the VLSI chips being considered.

The design of Fig. 7b and that of 9a have approximately the same number of pins. If this number of pins (due to the data path width) is near technological limits (and thus the design of Fig. 9b will not fit one chip), then the Generalized Cube interchange box is superior due to the logic/pin ratio. Assuming the cost of two chips with the same number of pins is about the same, an ADM network would be more than twice as expensive as a Generalized Cube network of the same size (when realized with these two respective chips). The logic/pin ratio of the ADM chip (Fig. 9a) can be improved considerably by implementing extra capabilities the ADM network is known to support [23, 24]. These capabilities include dynamic rerouting of blocked messages and stage look-ahead with rerouting for blockage prediction. None of the additional features requires any extra pins. The additional capabilities are possible because of the extra paths between input and output and thus are *not* available for the Generalized Cube network.

As advances in packaging technology continue, the cost difference between the Generalized Cube and the ADM will narrow considerably until the ADM is more cost-effective. To see this, examine Fig. 6. The larger the number of switching elements (of the type in Fig. 9b), in the same stage, that can be placed on a single chip, the more intra-stage busses can be internalized. This reduces the I/O overhead of the extra links. If a whole stage can be placed on one chip, then the ADM network requires the same number of chips and connections between chips as the Generalized Cube network. The assumption here is that the chip circuit density is not sufficient to support a crossbar but it will accommodate more logic than one stage of a Generalized Cube requires. The ADM network's structure thus fills a gap between the cube type networks and crossbars. Until very large portions of an interconnection network can be placed on a single chip, it is clear that the ADM network will be more expensive to implement than the Generalized Cube, though the difference will continue to decline. Thus, it is important to determine the networks' cost-effectiveness. It has already been pointed out that the ADM's capabilities are a superset of the Generalized Cube's. Another factor that is becoming more important as the construction of enormous systems is considered, will be discussed in the next section: robustness or inherent fault tolerance.

V. Robustness: A Comparison Of Degradation Under Component Failure

In this section, the robustness of each network is measured by removing a single component (link or switch) and counting the number of input and output ports that are affected. An input port is considered affected if it cannot send a message to all output ports. An output port is considered affected if there is at least one input port from which it cannot receive messages. Since the number of ports affected varies with the location of the removed link, averages are computed.

The average number of affected ports is calculated for both implementations of each network. These calculations are performed using two different rules for counting affected ports. The first rule requires all I/O ports to be considered. Under this rule, it has been shown that some permutation connections can be routed around a faulty link in the ADM network, but this is not true in general [33]. The second rule allows "severely" affected ports to be disabled and thus not included in the count. This rule takes into account a practical system response to a network fault: the disabling of some components so that operation can continue, but in a degraded mode. This is feasible if the network is used for asynchronous communication by cooperating processors (MIMD mode). If the network is used in a synchronous mode to perform permutation connections (SIMD mode) disabling some components is not feasible. Thus the following analysis applies only to use in MIMD environments. The second rule is implemented as follows. Referring to the graphs in Figs. 3 and 5, if a straight (or horizontal) arc at level j is removed, then input port j and output port j are disabled. If links are equated with arcs, one pair of I/O ports is disabled. If switches are equated with arcs, since two straight arcs are included in each switching element (Figs. 7b and 9b), two pairs of I/O ports are disabled. Thus, in Figs. 1 and 6, the I/O ports whose addresses correspond to the output labels on a given switching element are disabled if that switching element fails.

The results using the first rule are shown in Table I and using the second rule are in Table II. The derivations

Table I: Average number of affected I/O ports in the Generalized Cube and ADM networks when links and switches are removed. All ports are considered. Node=switch implementation corresponds to Figs. 3 and 2. Arc=switch implementation corresponds to Figs. 1 and 6.

	Node = Switch		Arc = Switch	
Failure	Link	Switch	Link	Int. Box
Generalized Cube	$\frac{2N-2}{n}$	$\frac{4N-2}{n+1}$	$\frac{4N-2}{n+1}$	$\frac{4N-4}{n}$
ADM	$\frac{N+n-1}{3n}$	$\frac{2N+n}{n+1}$	$\frac{2N+n}{n+1}$	$\frac{2N+4n-4}{n}$
Cube/ADM	≥6	≥2	≥2	≥2

Table II: Average number of affected I/O ports in the Generalized Cube and ADM networks when links and switches are removed. Severely affected ports are disabled and not counted.

	Node = Switch		Arc = Switch	
Failure	Link	Switch	Link	Int. Box
Generalized Cube	$\frac{2N-n-2}{n}$	$\frac{2N-2n-2}{n+1}$	$\frac{2N-2n-2}{n+1}$	$\frac{4N-4n-4}{n}$
ADM	0	0	0	0

of all the results tabulated here are too lengthy to include, but can be found in [20]. As an example of how the values are calculated, consider the case of a link failure in the ADM network, implemented by equating nodes with switches. Assume the link is in stage i (see Fig. 2). If the link is non-straight, none of the I/O ports are affected since the routing tag scheme in [24] can dynamically reroute to another path that does not use that link. For example, a path from input 1 to output 7 is $+2^2$, $+2^1$, straight, which uses the $+2^1$ link in stage 1, between switches 5 and 7. If that link is bad, the message can route straight, -2^1, straight and avoid it. If the bad link is a straight link at level j, then there are 2^{n-j-1} input ports that cannot send a message to output port j (namely those ports whose low order i bits of their addresses agree with j's). In this case, output port j is the only output port that cannot receive messages from all input ports. This fact is a result of the properties derived in [24]. For example, suppose the straight link in stage 1, level 4 (in Fig. 2) is bad. Output 4 will be unable to receive messages from inputs 0 and 4. On the other hand, all the other output ports are unaffected. Even though the $+2^2$, straight, $+2^1$ path from input 0 to output 5 includes the bad straight link, a message can simply take the straight, -2^1, -2^0 path. The average number of I/O ports affected by a bad straight link is calculated as:

$$\frac{1}{n}\sum_{i=0}^{n-1}(2^{n-i-1}+1)=\frac{1}{n}\sum_{i=0}^{n-1}(2^i+1)=\frac{N+n-1}{n}.$$

Since the failure of a $+2^i$ or a -2^i link does not affect any I/O ports, if link failures are equally likely, then the average over all links is one third of the above value.

In Table I, the ratio of the average number of affected I/O ports in the Generalized Cube to those in the ADM is computed. Regardless of network size, in the implementation in which switches are equated with nodes, a link failure in the Generalized Cube network affects six times as many ports, on the average, as a link failure in the ADM. For all the remaining types of failures and implementations, the ratio is two.

The measurement using the first rule is a very conservative indication of the robustness of the ADM network. Table II shows that under the second rule, the ADM network is very robust. When one or two pairs of I/O ports are disabled after the failure of a link or a switching element, respectively, all remaining I/O ports are unaffected. A failure can eliminate one path between given unaffected input and output ports, but routing tag methods exist for avoiding such faults by using an alternate path (as in the example of routing from input 0 to output 5 above) [24]. (A method for improving the robustness of the Generalized Cube network by adding one extra stage has been explored in [2].)

This analysis assumes that the failure of one component is independent of the failure of any other component. If all or a large part of one stage is implemented on a single chip, this assumption may or may not be valid. If it is not, then the networks should be reanalyzed to account for the new failure pattern exhibited.

To exploit the robustness of any network, it is implicit that faults can be detected and diagnosed. Such capabilities have been investigated in [3, 14, 30, 34]. Dynamic fault avoidance has been discussed in [23, 24].

VI. Conclusions

This paper has examined two classes of multistage interconnection network for use in parallel/distributed systems; the cube type and the data manipulator type. This was done by comparing a representative network from each class: the Generalized Cube and the Augmented Data Manipulator (ADM). From a straightforward comparison, it appears that the ADM network is more costly, but more powerful. This paper has attempted to quantify the differences in implementation costs by considering comparable implementation models for both networks. Furthermore, using a graph model as a basis, a quantitative measure of comparative robustness was derived.

References

[1] G. B. Adams III and H. J. Siegel, "On the number of permutations performable by the augmented data manipulator network," *IEEE Trans. Comp.*, Vol. C-31, pp. 270-277, Apr. 1982.

[2] G. B. Adams III and H. J. Siegel, "The extra stage cube: a fault-tolerant interconnection network for supersystems," *IEEE Trans. Comp.*, Vol. C-31, pp. 443-454, May 1982.

[3] D. P. Agrawal, "Testing and fault-tolerance of multistage interconnection networks," *Computer*, Vol. 15, pp. 41-53, Apr. 1982.

[4] G. A. Anderson, and E. D. Jensen, "Computer interconnection structures: taxonomy, characteristics, and examples," *ACM Computing Surveys*, Vol. 7, pp. 197-213, Dec. 1975.

[5] G. H. Barnes and S. F. Lundstrom, "Design and validation of a connection network for many-processor multiprocessor systems," *Computer*, Vol. 14, pp. 31-41, Dec. 1981.

[6] K. E. Batcher, "STARAN parallel processor system hardware," *AFIPS 1974 Nat'l. Comp. Conf.*, May 1974, pp. 405-410.

[7] K. E. Batcher, "The flip network in STARAN," *1976 Int'l. Conf. Parallel Proc.*, Aug. 1976, pp. 65-71.

[8] F. Briggs, et. al., "PUMPS architecture for pattern analysis and image data-base management," *Proc. Pattern Rec. and Image Proc. Conf.*, Aug. 1981, pp. 387-398.

[9] L. Ciminiera and A. Serra, "Modular interconnection networks with asynchronous control," *14th Annual Hawaii Int'l. Conf. Sci.*, Jan. 1981, pp. 210-218.

[10] A. M. Despain and D. A. Patterson, "X-tree: a tree structured multi-processor computer architecture," *5th Annual Int'l. Symp. Comp. Arch.*, Apr. 1978, pp. 144-151.

[11] D. M. Dias and J. R. Jump, "Analysis and simulation of buffered delta networks," *IEEE Trans. Comp.*, Vol. C-30, pp. 273-282, Apr. 1981.

[12] T. Feng, "Data manipulating functions in parallel processors and their implementations," *IEEE Trans. Comp.*, Vol. C-23, pp. 309-318, Mar. 1974.

[13] T. Feng, "A survey of interconnection networks," *Computer*, Vol. 14, pp. 12-27, Dec. 1981.

[14] T. Feng and C. Wu, "Fault-diagnosis for a class of multistage interconnection networks," *IEEE Trans. Comp.*, Vol. C-30, pp. 743-758, Oct. 1981.

[15] G. R. Goke, G. J. Lipovski, "Banyan networks for partitioning multiprocessor systems," *1st Symp. Comp. Arch.*, Dec. 1973, pp. 21-28.

[16] T. Lang and H. S. Stone, "A shuffle-exchange network with simplified control," *IEEE Trans. Comp.*, Vol. C-25, pp. 55-65, Jan. 1976.

[17] D. H. Lawrie, "Access and alignment of data in an array processor," *IEEE Trans. Comp.*, Vol. C-24, pp. 1145-1155, Dec. 1975.

[18] M. Malek and W. W. Myre, "A description method of interconnection networks," *IEEE Tech. Com. Dist. Proc. Quarterly*, Vol. 1, Feb. 1981, pp. 1-6.

[19] W. C. McDonald, J. M. Williams, "The advanced data processing test bed," *Compsac*, pp. 346-351, Mar. 1978.

[20] R. J. McMillen, Ph.D. thesis, in preparation.

[21] R. J. McMillen, G. B. Adams III, and H. J. Siegel, "Performance and implementation of 4x4 switching nodes in an interconnection network for PASM," *1981 Int. Conf. Parallel Processing*, Aug. 1981, pp. 229-233.

[22] R. J. McMillen and H. J. Siegel, "The hybrid cube network," *Distributed Data Acquisition, Computing, and Control Symp.*, Dec. 1980, pp. 11-22.

[23] R. J. McMillen and H. J. Siegel, "Performance and fault tolerance improvements in the inverse augmented data manipulator network," *9th Annual Int'l. Symp. Comp. Arch.*, Apr. 1982, pp. 63-72.

[24] R. J. McMillen and H. J. Siegel, "Routing schemes for the augmented data manipulator network in an MIMD system," *IEEE Trans. Comp.*, to appear Dec. 1982.

[25] D. S. Parker and C. S. Raghavendra, "The Gamma network: a multiprocessor network with redundant paths," *9th Annual Int'l. Symp. Comp. Arch.*, Apr. 1982, pp 73-80.

[26] J. H. Patel, "Performance of processor-memory interconnections for multiprocessors," *IEEE Trans. Comp.*, Vol. C-30, pp. 771-780, Oct. 1981.

[27] M. C. Pease, "The indirect binary n-cube microprocessor array," *IEEE Trans. Comp.*, Vol. C-26, pp. 458-473, May 1977.

[28] D. K. Pradhan and K. L. Kodandapani, "A uniform representation of single-and multistage interconnection networks used in SIMD machines," *IEEE Trans. Comp.*, Vol. C-29, pp. 777-791, Sept. 1980.

[29] U. V. Premkumar, R. Kapur, M. Malek, G. J. Lipovski, P. Horne, "Design and implementation of the banyan interconnection network in TRAC," *AFIPS 1980 Nat'l. Comp. Conf.*, June 1980, pp. 643-653.

[30] B. D. Rathi and M. Malek, "Fault diagnosis of networks," *Distributed Data Acquisition, Computing, and Control Symp.*, Dec. 1980, pp. 110-119.

[31] M. C. Sejnowski, E. T. Upchurch, R. N. Kapur, D. P. S. Charlu, and G. J. Lipovski, "An overview of the Texas Reconfigurable Array Computer," *AFIPS 1980 Nat'l. Comp. Conf.*, June 1980, pp. 631-641.

[32] H. J. Siegel, "Interconnection networks for SIMD machines," *Computer*, Vol. 12, pp. 57-65, June 1979.

[33] H. J. Siegel, R. J. McMillen, "Using the Augmented Data Manipulator Network in PASM," *Computer*, Vol. 14, pp. 25-33, Feb. 1981.

[34] H. J. Siegel and R. J. McMillen, "The multistage cube: a versatile interconnection network," *Computer*, Vol. 14, pp. 65-76, Dec. 1981.

[35] H. J. Siegel, R. J. McMillen, P. T. Mueller, Jr., "A survey of interconnection methods for reconfigurable parallel processing systems," *AFIPS 1979 Nat'l. Comp. Conf.*, June 1979, pp. 529-542.

[36] H. J. Siegel, L. J. Siegel, F. C. Kemmerer, P. T. Mueller, Jr., H. E. Smalley, Jr., and S. D. Smith, "PASM: a partitionable SIMD/MIMD system for image processing and pattern recognition," *IEEE Trans. Comp.*, Vol. C-30, pp. 934-947, Dec. 1981.

[37] H. J. Siegel, S. D. Smith, "Study of multistage SIMD interconnection networks," *5th Annual Int'l. Symp. Comp. Arch.*, Apr. 1978, pp. 223-229.

[38] S. D. Smith, "LSI design considerations for multistage interconnection networks for parallel processing systems," *14th Annual Hawaii Int'l. Conf. Sys. Sci.*, Jan. 1981, pp. 219-227.

[39] R. J. Swan, S. H. Fuller and D. P. Siewiorek, "Cm*: a modular, multi-microprocessor," *AFIPS 1977 Nat'l. Comp. Conf.*, June 1977, pp. 637-644.

[40] K. J. Thurber, "Interconnection networks - a survey and assessment," *AFIPS 1974 Nat'l Comp. Conf.*, May 1974, pp. 909-919.

[41] A. R. Tripathi, G. J. Lipovski, "Packet switching in banyan networks," *6th Annual Int'l. Symp. Comp. Arch.*, Apr. 1979, pp. 160-167.

[42] L. C. Widdoes, Jr., "The Minerva multimicroprocessor," *3rd Annual Int'l. Symp. Comp. Arch.*, Jan. 1976, pp. 34-39.

[43] C. Wu, T. Feng, "On a class of multistage interconnection networks," *IEEE Trans. Comp.*, Vol. C-29, pp. 694-702, Aug. 1980.

[44] W. A. Wulf, C. G. Bell, "C.mmp--a multi-miniprocessor," *AFIPS 1972 Fall Joint Comp. Conf.*, Dec. 1972, pp. 765-777.